**REMARKS**

Applicant is in receipt of the Office Action mailed March 9, 2007. Claims 1, 3, 23, 26, 27, 28, 30, 32, 34, 36, and 38 have been amended. Claims 1, 3-7, 9-10, 12-24, 26-28, 30-32, 34-36, and 38-39 are pending in the case. Reconsideration of the present case is earnestly requested in light of the following remarks. The Remarks herein also summarize the discussion in the Telephone Interview conducted on May 25, 2007.

**Advisory Action**

The Advisory Action objected to the language of claim 1, specifically, the term "successively larger", and asserted that it is inappropriate to end a method claim that utilizes iteration without particularly specifying "an understandable exit to this iteration". The Advisory Action further asserted that claim 1 did not clearly express the idea that debugged program code is moved from the remaining portion of the program to the first portion of the program as the code is debugged.

Applicant has amended claim 1 (and others) to clarify the nature of the claimed invention, and respectfully submits that the claims as currently written clearly distinguish over the cited art, and are statutory. Applicant further respectfully submits that there is no statutory basis for the assertion that a particular exit condition for iterative methods must be included in a claim, and that such an exit condition is not required for novelty or utility. For example, Applicant notes that were the iterations to be performed until the program were 80% debugged, this would still be a useful and tangible result. Moreover, Applicant submits that since any exit strategy or condition may be used within the scope of the invention, specifying any particular exit strategy or condition in the claim is not only unnecessary, but would limit the claimed invention scope inappropriately. Applicant respectfully directs the Examiner's attention to MPEP 2173.04, which clearly states that breadth of invention scope is not indefiniteness, and clearly expresses the conditions under which claims may be rejected for undue breadth, none of which apply to the present case.

Applicant presents the below arguments with respect to the Final Office Action of March 9, 2007, in light of the amended claims. The arguments presented in the previous Office Action Response of April 26, 2007 is hereby incorporated by reference.

**Objections**

Claims 1, 23, 28, 32, and 36 were objected to for the phrase "the first portion of the program is a successively larger portion of the program". Applicant has amended these claims to clarify the invention scope, replacing the language objected to with language emphasizing that in each iteration, the user adjusts the respective sizes of the first portion of the program (comprising debugged program instructions) and the remaining portion (to be further debugged), i.e., adjusts the partition between these portions, and that for at least some of the iterations, the adjusting includes moving debugged program instructions from the remaining portion to the first portion to increase the size of the first portion of the program. In other words, after such iterations, a greater part of the program has been debugged, and may be deployed to the targeted programmable hardware element. Applicant thus respectfully submits that claim 1 as currently written clearly describes the means and results of the iterative method.

The Office Action asserts that such functionality "would never be construed as increasing in size in a automated and programmatic fashion via successive increment (emphasis added) as claimed [sic]", that "the limitation thus cannot be construed semantically as a concrete and repeatable step (i.e., whereby incremental occurs by succession) [sic]", and that "In order to provide a proper method claim (according to 35 USC. section 112, first, 35 USC section 101), there should be no action step that depends on the aleatory and/or unpredictable factor played by a human intervention as disclosed (e.g., *the user may, may be temporarily, may increase in size)*.".

Applicant respectfully disagrees. Applicant notes that the Examiner has cited text from the Specification, not the claims, and submits that the claims do not include such equivocating language ("may"). Applicant further respectfully notes that the Specification describes numerous embodiments in which various features may or may not be included; however, the independent claims as written are clearly directed to

embodiments where user input modifying the remaining portion of the program to debug the remaining portion of the program, and adjusting respective sizes of the first portion and the remaining portion based on the debugging *is* received, and where for one or more iterations the adjusting *does* include moving debugged program instructions from the remaining portion to the first portion to increase the size of the first portion of the program. In other words, the claims clearly state that user input is received adjusting the sizes of the two portions, and that for at least some of the iterations, the adjusting includes moving debugged program instructions from the remaining portion to the first portion to increase the size of the first portion of the program, i.e., to increase the size of the debugged portion of the program. Thus, there is no "unpredictable factor played by a human intervention" in the claims.

Applicant further notes that such moving of program instructions from the remaining portion to the first portion (or moving code from the first portion to the remaining portion), thereby making the first portion successively larger (or smaller), is supported by the Specification at least in p.41:4 – 9, and p.42:22 – p.44:4.

Claims 3, 26, 30, 34, and 38, were objected to for similar reasons, particularly, for the phrase "successively smaller portion of the program". Applicant has also amended dependent claims 3, 26, 30, 34, and 38, replacing the "successively smaller portion of the program" language with clear language that states that "for at least one or more other iterations, said adjusting comprises moving program instructions from the first portion to the remaining portion for further debugging, thereby decreasing the size of the first portion of the program". Applicant further respectfully submits that the Specification supports the subject matter of amended dependent claims 3, 26, 30, 34, and 38, at least in p.43:3 – 15.

Applicant thus respectfully requests removal of the objection to claims 1, 23, 28, 32, and 36, and claims 3, 26, 30, 34, and 38.


**Double Patenting Rejection**

Claim 6 was rejected under the judicially created doctrine of obviousness-type double patenting as being unpatentable over claim 85 of U.S. Patent No. 7,024,660

("'660"), specifically, that claim 85 of '660 anticipates the features and limitations of claim 6.

In the Final Office Action, the Examiner states that the *repeating* step of amended claim 1 would overcome the double patenting rejection of claim 6, but for the above objection for improper language use. Applicant has addressed the objection above, and thus requests removal of the double patenting rejection.

## Section 102 Rejections

Claims 1, 3-7, 9-10, 12-24, 26-28, 30-32, 34-36, and 39-39 were rejected under 35 U.S.C. 102(b) as being anticipated by Tseng et al. (USPN: 6,009,256, "Tseng"). Applicant respectfully disagrees.

As noted above, anticipation requires the presence in a single prior art reference disclosure of each and every element of the claimed invention, arranged as in the claim. *Lindemann Maschinenfabrik GmbH v. American Hoist & Derrick Co.*, 221 USPQ 481, 485 (Fed. Cir. 1984). The identical invention must be shown in as complete detail as is contained in the claims. *Richardson v. Suzuki Motor Co.*, 9 USPQ2d 1913, 1920 (Fed. Cir. 1989).

Amended claim 1 recites:

1.    A memory medium comprising program instructions for debugging a program, wherein the program is intended for deployment on a programmable hardware element to perform a function, wherein the program instructions are executable to perform:

a) converting a first portion of the program into a first hardware configuration program which is deployable on the programmable hardware element to perform a corresponding first portion of the function, wherein the first portion of the program comprises debugged program instructions, and wherein a remaining portion of the program is to be debugged by a user;

b) configuring the programmable hardware element with the first hardware configuration program;

c) executing the program, wherein said executing comprises:

the programmable hardware element executing the first portion of the program; and

the computer system executing the remaining portion of the program;

wherein the remaining portion of the program is operable to be analyzed and debugged in response to said executing;

d) receiving user input modifying the remaining portion of the program to debug the remaining portion of the program, and adjusting respective sizes of the first portion and the remaining portion based on the debugging; and

repeating a) – d) one or more times in an iterative manner to debug the program, wherein for one or more iterations, said adjusting comprises moving debugged program instructions from the remaining portion to the first portion to increase the size of the first portion of the program.

Applicant respectfully submits that the Examiner has mischaracterized the cited art of Tseng, and has improperly read into Tseng various novel features and limitations of Applicant's claim 1, absent any evidence that Tseng actually discloses these features. In the previous Response, which is hereby incorporated by reference, Applicant provided arguments explaining numerous distinctions between Tseng and Applicant's invention as recited in claim 1, in response to which the Examiner has provided additional arguments. Applicant presents the following remarks, recapitulating previous arguments, and further addressing the Examiner's new arguments.

Applicant submits that Tseng nowhere teaches or suggests **d) receiving user input modifying the remaining portion of the program to debug the remaining portion of the program, and adjusting respective sizes of the first portion and the remaining portion based on the debugging**, nor **repeating a) – d) one or more times in an iterative manner, wherein for one or more iterations, said adjusting comprises moving debugged program instructions from the remaining portion to the first portion to increase the size of the first portion of the program**, as recited in amended claim 1.

Nowhere does Tseng disclose performing such converting, configuring, executing, and receiving user input modifying the remaining portion of the program and adjusting respective sizes of the first portion and the remaining portion based on the debugging, in an iterative fashion, where for at least some of the iterations, the adjusting includes moving debugged program instructions from the remaining portion to the first portion to increase the size of the first portion of the program, i.e., the debugged portion of the program. In other words, Tseng fails to teach or suggest such iterative debugging, where, as the program is debugged, the user moved debugged code from the remaining portion to the first portion, such that the (first) portion of the program that is deployed to the programmable hardware element increases in size, and thus, the remaining portion (executed by the computer system and debugged by the user) decreases in size. Said another way, in Applicant's invention as represented in claim 1, and as described in detail in the Specification, during the iterative debugging process, as the program is debugged, the debugged portions are moved from software implementation to hardware implementation, i.e., from the remaining portion to the first portion, thereby increasing the size of the first portion. Dependent claim 4 expresses the final result of such iterative incremental migration to the hardware portion (the first portion) from the software portion (the remaining portion), where, after the debugging process is complete, the entire program is converted and deployed to the programmable hardware element.

Applicant has reviewed cited Figures 2, 5, col. 15:34-57, and col.35:65 – col.36:50 carefully, and can find no description of these features. For example, neither Figure 2 nor Figure 5 discloses the claimed iterative debugging where debugged program instructions are moved from the remaining portion to the portion deployed to hardware, thereby making the first portion increase in size, i.e., where portions of the program are successively moved from software emulation to hardware deployment as they are debugged. Rather, these figures (and accompanying text) disclose switching back and forth between a software model and a hardware model to debug the circuit design, where the software model and the hardware model remain as originally defined or partitioned (see Figure 4 and accompanying text).

Neither figure discloses successively changing the sizes of the first portion and the remaining portion, i.e., successively changing the partition between the portion of the

program executing on the computer system and the portion of the program implemented in hardware. Rather, as Tseng makes clear, the user partitions the program once (again, see Figure 4 and corresponding text), then switches between software simulation and hardware emulation as desired. This point is clearly made in col.16:55-62:

> The reconfigurable hardware boards 250 contain the user's emulated circuit design. This SEmulation system has the ability to let the user selectively switch between software simulation and hardware emulation, as well as stop either the simulation or emulation process at any time, cycle-by-cycle, to inspect values from every component in the model, whether register or combinational.

Cited col. 15:34-57 describes emulating the circuit in the context of its target system environment. While this text does mention running the emulation multiple times to debug the circuit design, the text (and Tseng in general) fails to disclose moving (debugged) program instructions of the program from software emulation (the remaining portion) to hardware deployment (the first portion) as they are debugged, i.e., increasing the size of the portion deployed to the programmable hardware element. Rather, as the text describes, "After the emulation with the target system, the user has results that validate the circuit design or reveal non-functional aspects. At this point, the user can simulate/emulate again as indicated at step 135, stop altogether to modify the circuit design, or proceed to integrated circuit fabrication based on the validated circuit design."

Col.35:65 – col.36:50 describes logging operations for Tseng's debugging process, and makes no mention of iterative debugging where the portion of the program deployed to hardware is made successively larger as the program is debugged, i.e., where the program is incrementally moved from software emulation to hardware deployment during debugging.

Thus, based on the Examiner's asserted equivalence of Tseng's hardware and software models as the first portion and remaining portion of claim 1, nowhere does the cited text or Figures (or Tseng in general) disclose iteratively modifying the partition between the software model and the hardware model, where for one or more iterations, the user moves debugged program instructions from the remaining portion to the first portion to increase the size of the first portion of the program (and thus decrease the size of the software model, i.e., the remaining portion).

20

The Examiner asserts that Tseng's iterations teach Applicant's claimed repeating, stating that Applicant's claimed limitation that "for one or more iterations the first portion of the program is a successively larger portion of the program" has not been given significant weight because of the Examiner's Objection to the claim language. However, these issues have been resolved above, and so Applicant respectfully submits that this subject matter of claim 1, reworded per the present amendments, should be given its full weight.

Thus, Applicant respectfully submits that Tseng nowhere discloses moving debugged program instructions from the remaining portion (the software portion) to the first portion (the hardware portion) during the debugging process; rather, once the software model and the hardware model have been defined, they remain as originally defined or partitioned. In other words, the portion modeled in software, which in some cases may be the entire circuit, and the portion of the circuit modeled in hardware, e.g., for hardware acceleration, remains the same throughout the debugging process.

Tseng describes four modes of operation, an analysis of which was provided in the previous Response. As noted in that Response, in cases where Tseng's model is partitioned between software and hardware, it remains so, and so is not "intended for deployment on a programmable hardware element to perform a function". In cases where the circuit is only modeled in hardware (e.g., mode D), there is no "remaining portion". Thus, none of the disclosed modes of Tseng's system and method anticipate the features and limitations of claim 1. Moreover, even in combination, these disclosed modes of operation fail to teach Applicant's invention as represented in claim 1, at least for the reason that Tseng nowhere teaches iteratively increasing the (debugged) portion of the program deployed to a programmable hardware element during debugging, i.e., by moving debugged program instructions from the remaining portion (software portion) to the first portion (hardware portion).

The Examiner notes that Tseng allows the user to change register values for a rerun of the simulation/emulation, and asserts that this modifies the "first portion", and that this "reads on remaining portion relative to the first portion". Applicant does not understand what the Examiner means. Tseng's "register components are simple storage components", as Tseng states in col.17:42, and are typically part of the hardware model.

Applicant respectfully submits that changing a value in a register is not equivalent to Applicant's claimed modifying the remaining portion of the program to debug the remaining portion of the program, and adjusting respective sizes of the first portion and the remaining portion based on the debugging, and particularly where the adjusting includes moving debugged program instructions from the remaining portion to the first portion to increase the size of the first portion of the program, as claimed.

Thus, for at least the reasons provided above, Applicant respectfully submits that Tseng neither teaches nor suggests all the features and limitations of claim 1, and so claim 1 and those claims dependent therefrom are patentably distinct and non-obvious over the cited art, and are thus allowable.

Independent claims 23, 28, 32, and 36 include similar limitations as claim 1, and so the above arguments apply with equal force to these claims. Thus, for at least the reasons provided above, claims 23, 28, 32, and 36, and those claims respectively dependent therefrom, are patentably distinct and non-obvious over the cited art, and are thus allowable.

Independent claim 27 also recites limitations similar to those of claim 1, although in a slightly different form. More specifically, claim 27 explicitly recites two iterations of the iterative debugging process, where, as in claim 1, the program is partitioned into two portions, a first portion directed to a programmable hardware element, and comprising debugged program instructions, and, and a first remaining portion directed to a computer system for debugging by the user. The program is executed, including the programmable hardware element executing the first portion of the program, and the computer system executing the first remaining portion of the program. The remaining portion of the program is analyzed and debugged, e.g., by the user. After the user has debugged (modified) the remaining portion (in response to executing the portions on the programmable hardware element and computer system, respectively), the user repartitions the program, specifying a second portion for deployment on the programmable hardware element that includes the first portion and a debugged portion of the first remaining portion of the program, and a second remaining portion that includes only a subset of the first remaining portion of the program. The second portion is then converted and deployed to the programmable hardware element, and the program

executed, where the second portion is executed on the programmable hardware element, and the second remaining portion is executed by the computer system.

Nowhere does Tseng teach or suggest these features and limitations.

The Examiner again relies on the Objection to Applicant's claim language "for one or more iterations the first portion of the program is a successively larger portion of the program" in the rejection of claim 27. However, claim 27 does not use this language. Rather, as explained above, claim 27 explicitly describes the movement of a debugged program portion from the remaining portion of the program to the first portion of the program (and thus, from software simulation/emulation to hardware implementation) during the debugging process. Applicant respectfully notes that claim 27 has been amended to clarify that the first portion of the program comprises debugged program instructions. Thus, the Objection, directed to claims 1, 23, 28, 32, and 36, addressed above, is not germane to claim 27.

As argued at length above, Tseng nowhere discloses this feature (moving debugged portions of the program from software to hardware).

The Examiner provides no arguments against the patentability of claim 27 directly, but refers instead to arguments presented regarding claims 2 and 3. However, claim 2 has been cancelled, and amended claim 3 recites the limitation "wherein for one or more other iterations, said adjusting comprises moving program instructions from the first portion to the remaining portion for further debugging, thereby decreasing the size of the first portion of the program", which is not at all equivalent to the subject matter of claim 27, since claim 3 is directed to iterations in which a part of the program is moved *back* to the remaining portion from the first portion, e.g., for further debugging. Thus, the Examiner's rejection of claim 27 is unfounded, incorrect, and improper.

Thus, for at least the reasons provided above, Applicant respectfully submits that Tseng neither teaches nor suggests all the features and limitations of claim 27, and so claim 27 and those claims dependent therefrom are patentably distinct and non-obvious over the cited art, and are thus allowable.

Removal of the section 102 rejection of claim 1, 3-7, 9-10, 12-24, 26-28, 30-32, 34-36, and 38-39 is earnestly requested.

Applicant also asserts that numerous ones of the dependent claims recite further distinctions over the cited art. However, since the independent claims have been shown to be patentably distinct, a further discussion of the dependent claims is not necessary at this time.

**CONCLUSION**

Applicant submits the application is in condition for allowance, and an early notice to that effect is requested.

If any extensions of time (under 37 C.F.R. § 1.136) are necessary to prevent the above-referenced application(s) from becoming abandoned, Applicant(s) hereby petition for such extensions. The Commissioner is hereby authorized to charge any fees which may be required or credit any overpayment to Meyertons, Hood, Kivlin, Kowert & Goetzel P.C., Deposit Account No. 50-1505/5150-79600/JCH.

Also filed herewith are the following items:

☒ Request for Continued Examination

☐ Terminal Disclaimer

☐ Power of Attorney By Assignee and Revocation of Previous Powers

☐ Notice of Change of Address

☐ Other:

<div style="text-align: right;">

Respectfully submitted,

/Jeffrey C. Hood/
Jeffrey C. Hood, Reg. #35198
ATTORNEY FOR APPLICANT(S)

</div>

Meyertons, Hood, Kivlin, Kowert & Goetzel PC
P.O. Box 398
Austin, TX 78767-0398
Phone: (512) 853-8800
Date: June 8, 2007    JCH/MSW